CLAIMS

1. A method for modelling a non-linear transfer function with a power law function, the method comprising:
receiving a transfer function; and
iteratively, until a termination flag is set:
receiving a first power law function;
generating an auxiliary function from the transfer function and local differences between the transfer function and the first power law function;
fitting a second power law function to the auxiliary function;
calculating a modelling error from the second power law function and the transfer function; and
setting the termination flag when the modelling error is less than a predetermined value.

2. The method of claim 1, wherein:
receiving the first power law function in a given iteration comprises receiving the second power law function that was generated in the preceding iteration.

3. The method of claim 1, wherein:
receiving the first power law function in the first iteration comprises receiving a power law function generated by fitting the transfer function.

4. The method of claim 1, further comprising:
counting the number of iterations; and
setting the termination flag when the number of iterations exceeds a maximum number of iterations.

5. The method of claim 1, wherein:
the transfer function is a transfer function for gamma correction, and the first

and second power law functions are power law functions having a form of $c\,x^{\beta}$, wherein x is the input variable of the power law functions, and c and β are real numbers.

6. The method of claim 5, wherein:

fitting the second power law function to the auxiliary function includes fitting a linear function to a logarithmic representation of the auxiliary function.

7. The method of claim 6, wherein:

fitting the linear function to the logarithmic representation of the auxiliary function includes minimizing a least square error between the linear function and the logarithmic representation of the auxiliary function.

8. The method of claim 1, further comprising:

using a modifying parameter to weight the local differences between the transfer function and the first power law function that are used to generate the auxiliary function.

9. The method of claim 8, further comprising:

optimizing the modifying parameter.

10. The method of claim 9, wherein optimizing the modifying parameter comprises :

generating a plurality of auxiliary functions, wherein each auxiliary function is generated by weighting the local differences between the transfer function and the first power law function using a corresponding modifying parameter;

fitting each auxiliary function in the plurality of auxiliary functions to generate a plurality of second power law functions;

calculating a modelling error for each power law function in the plurality of second power law functions to generate a plurality of modelling errors; and

determining an optimal modifying parameter from the plurality of modelling errors.

11. The method of claim 10, wherein:

determining the optimal modifying parameter comprises determining a range of modifying parameters that includes the optimal modifying parameter.

12. The method of claim 10, wherein:

determining the optimal modifying parameter comprises performing a golden search for the modelling error that corresponds to the optimal modifying parameter.

13. The method of claim 1, wherein:

calculating the modelling error for the second power law function comprises calculating a total square error between the transfer function and the second power law function.

14. The method of claim 1, wherein:

calculating the modelling error for the second power law function comprises calculating the maximum absolute difference between the transfer function and the second power law function.

15. The method of claim 1, wherein:

receiving a transfer function comprises receiving a plurality of transfer function values.

16. The method of claim 1, wherein:

receiving a transfer function comprises receiving a piecewise continuous monotonically increasing transfer function.

17. A method for modelling a non-linear transfer function with a power law function, the method comprising:

receiving a transfer function;

fitting the transfer function with an approximating power law function; and

iteratively, until a termination flag is set:

reflecting the approximating power law function about the transfer function to generate an auxiliary function;

fitting the auxiliary function with a new approximating power law function;

calculating a modelling error from the new approximating power law function and the transfer function; and

setting the termination flag when the modelling error is less than a predetermined value.

18. A software product, tangibly embodied in an information carrier, for modelling a non-linear transfer function with a power law function, the software product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

receiving a transfer function; and

iteratively, until a termination flag is set:

receiving a first power law function;

generating an auxiliary function from the transfer function and local differences between the transfer function and the first power law function;

fitting a second power law function to the auxiliary function;

calculating a modelling error from the second power law function and the transfer function; and

setting the termination flag when the modelling error is less than a predetermined value.

19. The software product of claim 18, wherein:

receiving the first power law function in a given iteration comprises receiving the second power law function that was generated in the preceding iteration.

20. The software product of claim 18, wherein:

receiving the first power law function in the first iteration comprises receiving a power law function generated by fitting the transfer function.

21. The software product of claim 18, further comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

counting the number of iterations; and

setting the termination flag when the number of iterations exceeds a maximum number of iterations.

22. The software product of claim 18, wherein:

the transfer function is a transfer function for gamma correction, and the first and second power law functions are power law functions having a form of $c\,x^{\beta}$, wherein x is the input variable of the power law functions, and c and β are real numbers.

23. The software product of claim 22, wherein:

fitting the second power law function to the auxiliary function includes fitting a linear function to a logarithmic representation of the auxiliary function.

24. The software product of claim 23, wherein:

fitting the linear function to the logarithmic representation of the auxiliary function includes minimizing a least square error between the linear function and the logarithmic representation of the auxiliary function.

25. The software product of claim 18, further comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

using a modifying parameter to weight the local differences between the transfer function and the first power law function that are used to generate the auxiliary function.

26. The software product of claim 25, further comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

optimizing the modifying parameter.

27. The software product of claim 26, wherein optimizing the modifying parameter comprises :

generating a plurality of auxiliary functions, wherein each auxiliary function is generated by weighting the local differences between the transfer function and the first power law function using a corresponding modifying parameter;

fitting each auxiliary function in the plurality of auxiliary functions to generate a plurality of second power law functions;

calculating a modelling error for each power law function in the plurality of second power law functions to generate a plurality of modelling errors; and

determining an optimal modifying parameter from the plurality of modelling errors.

28. The software product of claim 27, wherein:

determining the optimal modifying parameter comprises determining a range of modifying parameters that includes the optimal modifying parameter.

29. The software product of claim 27, wherein:

determining the optimal modifying parameter comprises performing a golden search for the modelling error that corresponds to the optimal modifying parameter.

30. The software product of claim 18, wherein:

calculating the modelling error for the second power law function comprises calculating a total square error between the transfer function and the second power law function.

31. The software product of claim 18, wherein:

calculating the modelling error for the second power law function comprises calculating the maximum absolute difference between the transfer function and the second power law function.

32. The software product of claim 18, wherein:

receiving a transfer function comprises receiving a plurality of transfer function values.

33. The software product of claim 18, wherein:

receiving a transfer function comprises receiving a piecewise continuous monotonically increasing transfer function.

34. A software product, tangibly embodied in an information carrier, for modelling a non-linear transfer function with a power law function, the software product comprising instructions operable to cause one or more data processing apparatus to perform operations comprising:

receiving a transfer function;

fitting the transfer function with an approximating power law function; and

iteratively, until a termination flag is set:

reflecting the approximating power law function about the transfer function to generate an auxiliary function;

fitting the auxiliary function with a new approximating power law function;

calculating a modelling error from the new approximating power law function and the transfer function; and

setting the termination flag when the modelling error is less than a predetermined value.